```
 SSSSSS      IIIII     NN      N
S              I       N N     N
S              I       N  N    N
 SSSSS         I       N   N   N
      S        I       N    N  N
      S        I       N     NN
SSSSSS       IIIII     N       N
```

-Sinclair Information Nouvelles-

No.2 septembre 1985 $ 2.00

Rédacteur: Réal Gagnon (544 Hermine #2 Quebec P.Q. G1N 2G6)
Ce bulletin est conçu avec l'aide de traitement de texte
TASWORD II de Tasman Software (version TS2068)
et d'une imprimante SMITH-CORONA, Fastext 80.

** DANS CE NUMERO **

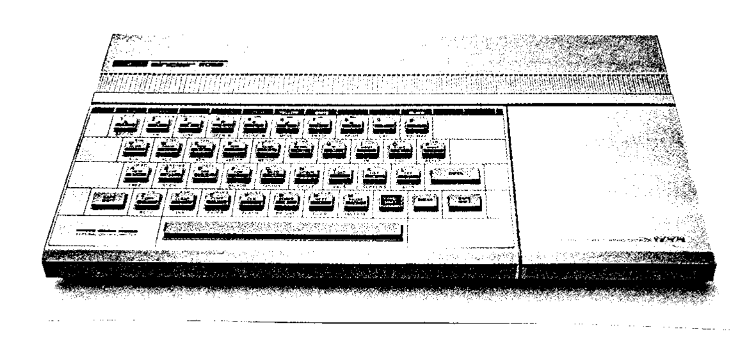
*PROGRAMME : GENERATEUR DE CARACTERE,redéfinir complètement le set de caractères du 2068 et maintenant chose facile

*PROGRAMME : LDIR,explorer les possibilités de cette instruction propre au Z80.

*PROGRAMME : TOOLKIT ULTRARES 2068,exploiter l' "EXTENDED COLOR MODE" qui était inaccessible jusqu'à ce jour.

*TECHNIQUE : Simuler les commandes SOUND et STICK avec la ROM du SPECTRUM.

*PLUS des TRUCS et ASTUCES,l'actualité,etc...





DERNIERE MINUTE
Le QL est maintenant disponible chez GAGNON ELECTRONIQUE, au prix de 799.95$.

** ACTUALITE SINCLAIR **

-L'ENIGMATIQUE QL

Au moment où sont écrites ces lignes le QL n'est pas encore disponible sur le marché nord-américain. Le distributeur officiel du QL pour le CANADA est EDG Electronics Distributors Inc.,3950 Chesswood Drive,Downsview ONTARIO,M3J 2W5. Le prix annoncé est $799.95 canadien. Le prix américain est a $499US.

Selon M. Yves Gagnon, le sympathique propriètaire de GAGNON ELECTRONIQUE,le QL devrait ètre disponible d'un moment à l'autre Pour plus d'information telephonez lui à 527-6103.

-LOGICIEL 2068

Si vous cherchez des logiciels pour votre 2068 non-modifié,vous pouvez écrire à E. MCGHEE,suite 557-21,10405 Jasper Avenue,Edmonton ALBERTA. On vous enverra un catalogue.

-EXTENSION MEMOIRE POUR LE 2068

Il est possible d'utiliser un module 16K du ZX81 sur un 2068. Ceci gràce au systeme de BANK SWITCHING du 2068. Pour plus d'information allez faire un tour chez GAGNON ELECTRONIQUE,390 Blvd. Central Nord à DUBERGER.

-L'ATTAQUE DES 32 BITS

Commodore vient d'annoncer la sortie de son 32 Bits,l'AMIGA. Avec ses 256K de RAM extensible a 4 MEG,un clavier de 86 touches ,un souris à 2 boutons,une résolution de 1024x800,l'AMIGA devient un concurrent sérieux a l'ATARI 520 ST et au QL. Son prix serait dans les $2000 US contre $799 pour le 520 ST et $499 US pour le QL. Qui sera le vainqueur ?

SIN

TRUC 2068...TRUC 2068...TRUC 2068...TRUC 2068...TRUC 2068...TRUC

-Si vous possèdez le logiciel TASWORD II pour le 2068,voici quelque POKEs utiles:

*Pour changer les couleurs de l'écran :
POKE 58512,24:POKE 58513,C:POKE 58521,54:POKE 58522,C
ou C égale à (8*PAPER)+INK.

*Pour la marge:POKE 58508,54:POKE 58509,C ->C=(8*PAPER)+INK

*Pour la bordure:POKE 64516,b -> b=couleur

*Pour les 2 lignes du bas: POKE 64570,C POKE 59993,C
C=(8*PAPER)+INK

(source SINCLAIR USER mai 85)

GENERATEUR de CARACTERES

Ce programme permet de redéfinir complètement le jeu de caractères du TS2068,soit des codes ASCII 32 a 122. Pour ce faire,il y à 2 méthodes,soit décimal ou binaire. Le programme est du type "MENU DRIVEN",i.e que toute l'information nécessaire pour bien utilisé le GENERATEUR est incluse dans des menus , vous n'avez qu'à choisir une des options proposées.

OPTION #1 -Cette option vous permet de compléter une police de caractères que vous n'auriez pas eu le temps de terminer.

OPTION #2 -Permet une vue d'ensemble du travail accompli.

OPTION #3 -C'est içi que vous créez vos propres caractères. En mode décimal,vous donnez une nouvelle valeur à chacun des 8 octets qui forme le caractère. En mode binaire, vous déplacer le curseur à l'intérieur d'une grille, la touche -1- laisse le pixel à l'état ON (INK) et la touche -0- le laisse à l'état OFF (PAPER).

OPTION #4-5-6 - Les mêmes que 1-2-3 sauf que içi,c'est les UDGs.

OPTION #7 -Sauvegarde sur cassette du travail accompli. Les directives pour l'utilisation future de vos nouveaux caractères vous seront données alors.

OPTION #8 -Cette option vous permet d'agencer vos caractères ou UDGs comme bon il vous semble et vérifier à l'écran si le "nouveau LOOK" de vos caractères ou UDGs vous plait.

OPTION #9 -Efface tous vos nouveaux caractères et recopie les caractères SINCLAIR standards.

OPTION #10-Fin du programme. La commande NEW s'effectue, à noter que vos caractères sont toujours en mémoire.

Votre set de caractères se trouve a l'adresse 64000.Pour l'utiliser nous devons faire pointer la variable CHARS à cette adresse ,faites: POKE 23606,0 et POKE 23607,249. Pour revenir au caractères standards,faites: POKE 23606,0 et POKE 23607,60.

Les datas de la ligne 1500 sont une routine en langage machine qui copie très rapidement les caractères SINCLAIR dans la RAM. Veuillez noter que à la ligne 2500 il s'agit bien du TOKEN BIN. Ne pas écrire BIN lettre par lettre.

Liste: GENERATEUR de CARACTERES   GAG-o84   TS 2068

```
  1 REM ________________________________GENERATEUR de CARACTERES    GAG-o 1984
  2 CLEAR 63000
 10 GO SUB 9500
 15  BORDER 0: PAPER 0: INK 6: CLS
 16 LET o=0: LET in=0: LET copie=0
 20 REM ________________________________MENU
 25 GO SUB 150
 30 CLS : PRINT  AT 0,4; INVERSE 1;"GENERATEUR de CARACTERES"' AT 1,9;"GAGNON-o 1984"'' INVERSE 0;" 1-Charge un set de cara
cteres"'" 2-Vue du set de caracteres"'" 3-Modification d'un caracteres"'" 4-Charge un set d'UDGs"
 35 PRINT " 5-Vue des UDGs"'" 6-Modification d'un UDGs"'" 7-Sauve caracteres/UDGs"'" 8-Tableau d'essai"'" 9-Efface vos cara
cteres"'"10-Fin du programme"
 40 PRINT '' TAB 2;"N.B. -UDG- signifie "'"      USER DEFINED GRAPHIC."
 45 INPUT "Votre choix (1-8):";i
 50 IF ((i<1)+(i>10)) THEN  GO TO 45
 55 BEEP .1,3*i: FOR a=0 TO 7: BEEP a/100,60: PRINT  AT i+2,0; OVER 1; PAPER a;"                                ": NEXT a
100 IF i=10 THEN  NEW
105 GO SUB 100+(100*i): GO TO 30
150 REM ________________________________CARACTERE SINCLAIR
155 POKE 23607,60: RETURN
160 REM ________________________________CARACTERE REDEFINI
165 POKE 23607,249: RETURN
200 REM ________________________________OPTION #1
205 LET copie= NOT copie
210 CLS : PRINT  AT 10,1;" CHARGE d'un set de CARACTERES"
215 GO SUB 2000: RETURN
300 REM ________________________________OPTION #2
305 CLS : PRINT  AT 0,3; INVERSE 1;" VUE du set de CARACTERES "
310 PRINT '''*Du code ASCII 32 a 127"''Sauf STICK et FREE"'''
315 IF  NOT copie THEN  GO SUB 9600
320 GO SUB 160
325 PRINT " ";: FOR i=1 TO 96 STEP 6: FOR j=0 TO 5: LET j$= CHR$ (31+j+i)
330 IF j$=" STICK " OR j$=" FREE " THEN  NEXT j
335 PRINT j$;"   ";: NEXT j: NEXT i
340 GO SUB 150: PRINT  AT 20,0;"Une touche...": PAUSE 0: RETURN
400 REM ________________________________OPTION #3
405 CLS : IF  NOT copie THEN  GO SUB 9600
410 PRINT  AT 0,7; INVERSE 1;"  MODIFICATION  "
415 INPUT "Quel caractere:"; LINE c$
420 PRINT '"Le caractere """;: GO SUB 160: PRINT c$;: GO SUB 150: PRINT """.....";: PRINT """";c$;"""''
425 IF  CODE c$=0 THEN  RETURN
430 LET k=144: GO SUB 2050: LET a=3: LET ii=0
435 FOR x=159 TO 152 STEP -1: LET a=a+1: PRINT  AT a,10;: FOR y=112 TO 119
440 GO SUB 2060
445 NEXT y: IF i=3 THEN  PRINT  OVER 1; INK 1;"  "; INK 6; PEEK (64000+(8*( CODE c$-32)+ii))
446 IF i=6 THEN  PRINT  OVER 1; INK 1;"  "; INK 6; PEEK ( USR c$+ii)
447 LET ii=ii+1: NEXT x
450 LET k=72: GO SUB 2050
453 LET l=13:
455 POKE 23658,0: INPUT "Binaire ou Decimal (b/d):"; LINE i$: IF i$="b" THEN  GO TO 2100
460 IF i$ <> "d" THEN  GO TO 455
462 FOR q=0 TO 7
465 INPUT "La valeur:"; LINE v$: IF v$="" THEN  GO TO 465
466 LET n= VAL v$
```

(suite page [illegible])

(suite de la page précédente)

```
470 IF n>255 THEN  GO TO 465
471 IF i=3 THEN  POKE (64000+8*( CODE c$-32)+q),n
472 IF i=6 THEN  POKE  USR c$+q,n
475 LET m=n: PRINT  AT 1,10;: GO SUB 2080
480 FOR w=1 TO 8: GO SUB 2070: NEXT w: PRINT  OVER 1; INK 1;"  "; INK 6;m: LET l=l+1: NEXT q
485 INPUT "": PRINT #0;"Le resultat est [ ";: GO SUB 160: PRINT #0;c$;: GO SUB 150: PRINT #0;" ] pour [ ";c$;" ]": PAUSE 32
0: GO TO (400 AND i=3)+(700 AND i=6)
500 REM _________________________________OPTION #4
505 CLS : PRINT  AT 10,9; INVERSE 1;" CHARGE  U D G "'':
510 GO SUB 2000: RETURN
600 REM _________________________________VUE des UDGs
605 CLS : PRINT  AT 0,9; INVERSE 1;" VUE des U D Gs "''''
610 FOR j=144 TO 153 STEP 2: PRINT "  "; CHR$ j;"  "; CHR$ (j+1);: NEXT j
615 PRINT : PRINT : FOR i=j TO 162 STEP 2: PRINT "  "; CHR$ i;"  "; CHR$ (i+1);: NEXT i: PRINT ''"  "; CHR$ 164: PRINT ''''
"Une touche...": PAUSE 0: RETURN
700 REM _________________________________MODIFICATION UDG
705 CLS : PRINT  AT 0,5; INVERSE 1;" MODIFICATION UDGs "
710 POKE 23658,8: INPUT "Quel UDG (A-U):"; LINE c$: LET o=0: IF  CODE c$=0 THEN  RETURN
715 IF c$<"A" OR c$>"U" THEN  GO TO 710
720 PRINT '"Le U D G    [ "; CHR$ ( CODE C$+79);" ].....[ ";c$;" ]"
725 LET c$= CHR$ ( CODE c$+79): GO TO 430
800 REM _________________________________SAUVE UDG/CAR
805 CLS : PRINT  TAB 12;" SAUVE "''''
810 PRINT "  1-Set de caracteres"''  2-UDGs"'"  3-Caracteres+UDGs"
815 INPUT "Votre choix:";c
820 IF c<1 OR c>3 THEN  GO TO 815
830 CLS : IF c=1 OR c=3 THEN  PRINT "Sauve les caractere :";: INPUT "Le nom:"; LINE n$: PRINT n$'"Pour utiliser ces caracte
res"'"dans vos programme:"''"CLEAR 63999:LOAD nom CODE"'"POKE 23606,0:POKE 23607,249": SAVE n$ CODE 64000,768: GO SUB 850
835 CLS : IF c <> 1 THEN  PRINT "Sauve les UDGs:";: INPUT "Le nom:"; LINE n$: PRINT n$''"Pour utiliser ces UDGs:"'"LOAD no
m CODE": SAVE n$ CODE  USR "a",168: GO SUB 850
840 RETURN
850 POKE 23658,0: INPUT "Verification (o/n):"; LINE v$: IF v$="o" THEN  PRINT '"Je verifie ";n$: VERIFY n$ CODE : PRINT ''
O.K.": PAUSE 50:
855 RETURN
900 REM _________________________________TABLEAU d'ESSAI
905 CLS : PRINT  AT 0,7; INVERSE 1;" TABLEAU D'ESSAI "
910 INPUT "(1)UDG  (2)caractere:";n
915 IF n<1 OR n>2 THEN  GO TO 910
920 PRINT  AT 1,10; INVERSE 1; BRIGHT 1;(" CARACTERE " AND n=2)+(" UDG:(A-U) " AND n=1)
925 PRINT ''"Faites ENTER pour Terminer...": GO SUB 160
926 IF n=1 THEN  POKE 23617,2
930 INPUT  AT 16,0; AT 0,0; LINE m$
935 POKE 23617,0: GO SUB 150: RETURN
1000 REM _________________________________RESET
1005 POKE 23657,0: CLS : INPUT  AT 15,0; AT 0,0;"ETES-VOUS SUR (o/n):"; LINE n$
1010 IF n$="o" THEN  RANDOMIZE  USR 63000: LET copie= NOT copie
1015 RETURN
1999 STOP
2000 REM _________________________________CHARGE
2005 INPUT "Le nom:"; LINE a$
2010 PRINT ''"Je charge ";a$: LOAD a$ CODE
2015 PRINT ''"O.K. Une touche...": PAUSE 0: RETURN
2050 REM _________________________________GRILLE
2055 INK 1: PLOT 79,k: DRAW 65,0: DRAW 0,-65: DRAW -66,0: DRAW 0,65: INK 6: RETURN :
2060 REM _________________________________GRILLE BINAIRE
2065 PRINT  PAPER 7; INK 0;( CHR$ 143 AND  POINT (y,x))+(" " AND  NOT  POINT (y,x));: RETURN
```

(suite page suivante)

```
2070 PRINT  PAPER 7; INK 0;( CHR$ 143 AND b$(w)="1")+(" " AND b$(w)="0");: RETURN
2080 REM ________________________________DEC -> BIN
2085 LET b$="00000000": LET j=8
2090 LET n1=n: LET n= INT (n/2): LET b$(j)= STR$ (n1-n*2)
2095 IF n>0 THEN  LET j=j-1: GO TO 2090
2096 RETURN
2100 REM ________________________________CURSEUR BINAIRE
2105 DIM w$(8,8)
2110 IF  NOT in THEN  GO SUB 8000: INPUT ""
2115 LET x=12: LET y=9: LET a=1: LET b=1: PRINT #0;"Allez lentement avec le curseur."
2116 PRINT  AT x+a,y+b; OVER 0;"_"
2120 LET a$= INKEY$
2122 IF a$="" THEN  GO TO 2120
2136 BEEP .0001,60:
2140 IF a$="1" OR a$="0" THEN  PRINT  INK 0; PAPER 7; AT x+a,y+b;( CHR$ 143 AND a$="1")+(" " AND a$="0"): LET w$(a,b)=a$: LE
T b=b+1
2141 LET a=a+((a$="6")*(a<8))-((a$="7")*(a>1)): LET b=b+((a$="8")*(b<8))-((a$="5")*(b>1))
2145 IF a=8 AND b=9 THEN  GO TO 2500
2146 IF b=9 THEN  LET b=1: LET a=a+1
2160 GO TO 2116
2500 LET a$=" BIN "
2505 LET l=13: FOR q=0 TO 7
2510 LET x= VAL (a$+w$(q+1))
2515 PRINT  AT l,21; INK 1;"  "; INK 6;x: LET l=l+1
2520 IF i=3 THEN  POKE (64000+8*( CODE c$-32)+q),x
2530 IF i=6 THEN  POKE  USR c$+q,x
2535 NEXT q
2540 GO TO 485
8000 REM ________________________________MESSAGE
8005 LET in= NOT in
8010 LET m$="NOTE: Modification-Binaire,vous pouvez vous servir des touches 5 a 8 pour vous deplacer a l'interieur de la gri
lle. La touche -1- donne INK et la touche -0- donne PAPER."
8015: FOR q=31 TO 1 STEP -1: PAUSE 4: BEEP .0001,60: PRINT #0; BRIGHT 1; TAB q;m$( TO 32-q); AT 0,0: NEXT q
8020 FOR q=0 TO  LEN m$: PAUSE 4: BEEP .0001,64: IF 32+q> LEN m$ THEN  GO TO 8050:
8025 PRINT #0; BRIGHT 1;m$(q+1 TO 32+q); AT 0,0: NEXT q
8050 FOR q= LEN m$-32 TO  LEN m$: PAUSE 4: BEEP .0001,60: PRINT #0; BRIGHT 1;m$(q+1 TO  LEN m$); AT 0,0: NEXT q: RETURN
9500 REM ________________________________ROUTINE MACHINE
9510 FOR i=0 TO 11: READ a: POKE 63000+i,a: NEXT i: RETURN
9520 DATA 17,0,250,33,0,61,1,0,3,237,176,201
9600 REM ________________________________ROM -> RAM
9605 RANDOMIZE  USR 63000: LET copie= NOT copie: RETURN
9999 SAVE "car" LINE 1: VERIFY ""
```

2068...TRUCS2068...TRUCS2068...TRUCS2068...TRUCS2068...TRUCS2068

............Conversion BASIC MICROSOFT->BASIC SINCLAIR............

Commande LEFT$(A$,x) => DEF FN L$(A$,x)=A$(TO x)

Commande RIGHT$(A$,x)=> DEF FN R$(A$,x)=A$(x+1 TO)

Commande MID$(A$,x,t)=> DEF FN m$(A$,x,y)=A$(x TO (x+y)-1)

SOUND et STICK

Depuis la démission de Timex du domaine de la micro, plusieurs propriétaires de 2068 possèdent maintenant un SPECTRUM britannique. Ceci est possible par le remplacement de la ROM du 2068 par celle d'un SPECTRUM. Mais il y a un problème avec le BASIC du SPECTRUM. Il n'y a pas de FREE, ON ERR, SOUND ou de STICK.

Mais voici une bonne nouvelle: -Nous pouvons simuler certaines de ces commandes, et ce, avec le BASIC du SPECTRUM.
Dans le présent article, quand je parle du SPECTRUM 2068, je fais référence à un 2068 avec une ROM du SPECTRUM à l'intérieur.

1) SOUND.......................

Le 2068 possède un "chip" que le SPECTRUM n'a pas. Il s'agit d'un générateur de son, le GENERAL INSTRUMENT AY-3-2912. Il peut être programmer via le port F5h et le port F6h. Le premier sert à adresser un des 15 registres tandis que l'autre place une certaine valeur dans celui-çi.

Alors en BASIC 2068, une ligne de programme comme:

```
10 SOUND 8,13;7,62
```

peut devenir en SPECTRUM:

```
10 OUT 245,8:OUT 246,13:OUT 245,7:OUT 246,62
```

Comme vous pouvez le constater la conversion est très simple:

2068 BASIC → SOUND x,y;...;x,y

SPECTRUM BASIC → OUT 245,x:OUT 246,y:...:OUT 245,x:OUT 246,y

Si nous prenons l'exemple dans le manuel d'instruction a la page 195, la version pour le SPECTRUM 2068 sera:

```
 1 REM BOMBE SIFFLANTE
 5 FOR i=1 TO 2
10 READ registre,valeur
15 OUT 245,registre
16 OUT 246,valeur
17 NEXT i
20 DATA 7,62,8,15
30 OUT 245,0: FOR i=0 TO 100
35 OUT 246,i: PAUSE 2:NEXT i
50 REM EXPLOSION
55 FOR i=1 TO 10
60 READ registre ,valeur
65 IF registre=9999 THEN GOTO
   80
70 OUT 245,registre:OUT 246,val
   eur:NEXT i
75 DATA 6,6,7,7,8,16,9,16,10,16
   ,12,56,13,8,9999,0
80 PRINT "O.k.":PAUSE 1000
90 FOR i=8 TO 10:OUT 245,i:OUT
   246,0:NEXT i
```

La ligne 90 remet à zéro les trois registres qui contrôlent l'amplitude (volume).

Pour plus d'information sur la programmation de sons, je vous recommende le livre de F. Mazur, TIMEX-SINCLAIR 2068 INTERMEDIATE-ADVANCED GUIDE, publié chez SAMS BOOKS.

2) STICK.......................

Il y a 3 ports entrée/sortie qui sont dédiés a la lecture des JOYSTICKS . En premier lieu nous devons faire:

```
OUT 245,14
```

(suite page suivante)

(…te d [illegible])

Puis nous pouvons faire:

```
IN 4598 -> STICK 1 (gauche)
IN 4854 -> STICK 2 (droite)
IN 5110 -> STICK 1-2
```

Alons-y avec une démonstration:

```
 1 REM STICK SPECTRUM 2068
 5 PRINT " STICK 1";TAB 10;" S
TICK 2";TAB 20;"STICK 1-2"
 6 PRINT "IN 4598";TAB 10;"IN
4854";TAB 20;"IN 5110"
10 OUT 245,14
15 LET stick1=IN 4598
16 LET stick2=IN 4854
17 LET stick12=IN 5110
20 PRINT AT 3,2;stick1;TAB 12;
stick2;TAB 22;stick12
25 PRINT AT 4,3;255-stick1;TAB
13;255-stick2;TAB 23;255-stick
12:GOTO 10
```

A la ligne 25,nous enlevons de 255 la valeur retournée pour obtenir la même valeur retournée par la commande "STICK" en 2068 BASIC.

Lorsque le bouton "FEU" du joystick est pressé,la valeur retournée est 128,ceci est dûe à la bit 7 qui est mise à "1". Examinons cette valeur retournée sous sa forme binaire,nous remarquons ceci:

```
Bit 0   -si 0 -> HAUT
Bit 1   -si 0 -> BAS
Bit 2   -si 0 -> GAUCHE
Bit 3   -si 0 -> DROITE
Bit 4-6 -non-utilisées
Bit 7   -si 0 -> FEU
```

Pour une lecture rapide des ports JOYSTICKS,voici 3 routines en LANGAGE MACHINE.Les valeurs sont en décimales et sont entièrement relogeable.

```
STICK1=1,0,0,62,14,211,245,62,
       13,219,246,47,79,201
STICK2=1,0,0,62,14,211,245,219,
       246,47,79,201
STICK12=1,0,0,62,14,211,245,62,
       15,246,47,79,201
```

En ASSEMBLEUR,la lecture du STICK1 serait:

```
LD BC,0       ;registre BC=0
LD A,14
OUT (#F5),A   ;OUT 245,14
LD A,13
IN A,(#F6)    ;IN 4598
CPL           ;255-A
LD C,A        ;BC=255-A
RET           ;retour en BASIC
```

Nous pouvons "pokez" ces valeurs avec un petit programme comme:

```
 5 LET x=adresse
10 FOR i=0 TO data-1
20 READ valeur
30 POKE x+i,valeur
40 NEXT i
50 DATA placez ici les datas
de la routine de votre choix.
```

La ligne 10 signifie le nombre de datas moins 1.

Dans votre programme BASIC, vous appelez cette routine avec LET stick=USR adresse.

```
ex. 10 LET stick=USR 50000
    20 IF stick=1 THEN LET x=x+
       1
    30 IF stick=2 THEN LET x=x-
       1
    etc...
```

SIN

2068...TRUCS2068...TRUCS2068...

Pour avoir une ligne zero:
POKE 26711,0

Programme: L D I R

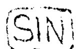
Ce programme permet de démontrer toute la puissance que l'instruction LDIR du Z80 contient. LDIR signifie "LoaD Increment & Repeat". Cette instruction, qui occupe seulement 2 octets, réalise en un temps incroyable plusieurs opérations de base.

Rappelez-vous que HL représente celui qui donne, DE celui qui reçoit et BC contient le nombre de fois que cette opération doit être faites. Donc LDIR fait

```
(HL)->(DE)    ;HL est chargé dans DE
DE=DE+1       ;DE est augmenté de 1
HL=HL+1       ;HL est augmenté de 1
BC=BC-1       ;BC est diminué de 1
BC >= 0       ;si BC est >0,on conti-
               nue,si BC=0 alors on
               arrête le transfert.
```

Tapez le programme qui suit et essayez les valeurs suivantes:

1) HL=0   DE=16384   BC=6000 ou BC=6912
2) HL=65368   DE=16384   BC=168
3) HL=26711   DE=16384   BC=6000




```
  0> REM _______________________________TRANSFERT de DONNEES   <--LDIR-->   par YVES GAGNON
  10 POKE 23658,8: PAPER 1: BORDER 1: INK 7: CLS
  30 PRINT "LDIR ASSEMBLEUR :"
  40 PRINT "================="
  45 PRINT '"Mnemoniques","Code Machine"
  46 PRINT "-----------","------------"
  50 DIM L(12): DIM A(3)
  60 LET L(1)=33: LET L(4)=17: LET L(7)=1: LET L(10)=237: LET L(11)=176: LET L(12)=201
 100 INPUT "ENTRER L'ADRESSE DU BLOC"'"D'OCTETS  A  TRANSFERRER"'"DANS  LE  REGISTRE HL  ? ";A(1)
 110 LET L(3)=A(1)/256: LET L(3)= INT L(3): LET L(2)=A(1)-(L(3)*256): GO SUB 500
 200 INPUT "ENTRER L'ADRESSE DU BLOC"'"D'OCTETS   A   RECEVOIR"'"DANS LE  REGISTRE DE   ? ";A(2)
 210 LET L(6)=A(2)/256: LET L(6) = INT L(6):LET L(5)=A(2)-(L(6)*256): GO SUB 510
 300 INPUT "ENTRER   LE    NOMBRE"'"D'OCTETS A TRANSFERER"'"DANS LE REGISTRE BC ? ";A(3)
 310 LET L(9)=A(3)/256: LET L(9)= INT L(9): LET L(8)=A(3)-(L(9)*256): GO TO 520
 500 PRINT '"LD  HL,";A(1),L(1);",";L(2);",";L(3): RETURN
 510 PRINT '"LD  DE,";A(2),L(4);",";L(5);",";L(6): RETURN
 520 PRINT '"LD  BC,";A(3),L(7);",";L(8);",";L(9)
 530 PRINT '"LDIR",L(10);",";L(11)
 540 PRINT '"RET",L(12)
1000 FOR M=1 TO 12: POKE (65520-1)+M,L(M): NEXT M
1500 PRINT #0;"<S> POUR SAUVEGARDE D'LA ROUTINE"'"<D> POUR DEBUT DU PROGRAMME"'"<E> POUR EXECUTER  LDIR"
1600 PAUSE 0
1610 IF  INKEY$ ="S" THEN  SAVE "LDIR" CODE 65520,12: PRINT #0;"Une touche...": PAUSE 0: RUN
1620 IF  INKEY$ ="D" THEN  RUN
1630 IF  INKEY$ ="E" THEN  RANDOMIZE  USR 65520: PRINT #0;"Une touche...": PAUSE 0: RUN
1640 GO TO 1600
9999 SAVE "LDIR" LINE 10: SAVE "LDIR" LINE 10: VERIFY "LDIR": VERIFY "LDIR"
```

PROGRAMME: TOOLKIT ULTRA RES 2068

Parmi les principales différences qui existent entre le SPECTRUM et le 2068, il y a les differents modes vidéos implantés dans le 2068. Mais ces différents modes ne sont pas accessibles par le BASIC du 2068. Le logiciel qui suit nous permet d'utiliser le mode appelé "EXTENDED COLOR MODE". Comme son nom l'indique, la section dévouée à la gestion de la couleur est multipliée par 8. Si en mode normal nous faisons: CIRCLE 125,87,80 et puis : PLOT 0,87: DRAW FLASH 1;255,0 nous verrons que l'attribut FLASH s'applique à chaque chaque bloc de 8 octets, alors que en mode "ECM" l'attribut s'applique à chaque octet.

Le TOOLKIT ne requiert aucun POKE pour opérer. Quand nous avons besoin de passer des informations aux routines machines, on peut le faire directement du BASIC selon la forme: INPUT USR adr, X,Y où adr est le point d'entrée de la routine, X et Y sont les 2 informations que la routine demande. Aussi inclus dans le TOOLKIT, quelques routines supplémentaires, que le programmeur averti sera apprecié, telle que FILL et SCROLL dans les 4 sens. Vous trouverez plus loin la liste complète du TOOLKIT en ASSEMBLEUR ainsi qu'un "HEXLOADER" qui se charge de "pokez" le TOOLKIT en mémoire et le sauvez par la suite.

Premièrement, tapez le HEXLOADER puis RUN 500 pour se faire une copie de securité sur cassette, au cas où. Après cela, faites RUN et attendez. Si jamais il y a une erreur dans les DATAs, le numéro de la ligne où s'est produite l'erreur sera affiché à l'écran. Si tout est en ordre, le programme se termine avec le message d'erreur "OUT OF DATA". Alors faites RUN 550 pour sauvez vos codes. Un RANDOMIZE USR 62000 rend actif l'EXTENDED COLOR MODE, un message de bienvenue devrait apparaitre au haut de l'écran, sinon faites PRINT USR 0 et rechargez le HEXLOADER pour vérification. Lors des utilisations futures du TOOLKIT, il est important de faire avant toute chose un CLEAR 61999 pour le protéger de toute "invasion exterieure".

Voici maintenant une description des possibilités du TOOLKIT. Et cela, en suivant leur ordre d'apparition dans la liste ASSEMBLEUR.

-RANDOMIZE USR 62000 , rend en fonction D_FILE2, qui est utilisé pour la gestion des couleurs. D_FILE2 débute à l'adresse 24576 et se termine à 30719. Le premier CALL met EXROM en fonction, ce sont les 8K du ROM suplémentaire que possèdons sur le SPECTRUM. Puis nous appelons dans le EXROM une routine qui se charge de déplacer le "FUNCTION DISPATCHER" et beaucoup d'autre chose. Le Troisième CALL remet le HOME ROM en place et renvoie EXROM d'où il vient.

-RANDOMIZE USR 62048 , efface D_FILE1 et D_FILE2, si vous faites le CLS du BASIC, seulement D_FILE1 sera effacé. RANDOMIZE USR 62051 efface seulement D_FILE2. Supposons que nous avons un dessin à l'ecran et que nous voulons changez le INK et le PAPER nous n'avons qu'à faire: PAPER X:INK Y:RANDOMIZE USR 62051 et les attributs sont changés sans effacés notre dessin, chose impossible avec le CLS BASIC. Donc pour effacer complètement l'écran faites RANDOMIZE USR 62048.

-INPUT USR 62069,X,Y , sert à mettre de la couleur sur un PLOT.
X égale à (8*PAPER)+INK, ainsi pour PAPER bleu et INK blanc, X=(8*1)+7=15. Y est pour FLASH et BRIGHT.
Y=1 ->BRIGHT 1, Y=2 ->FLASH 1, Y=3 ->BRIGHT 1 & FLASH 1
Y=0 ->BRIGHT 0 & FLASH 0.
ex. PLOT 0,0:INPUT USR 62067,56,2 met le point (0,0) à PAPER 7 et INK 0 avec FLASH 1.
Notez que ces 2 arguments ne pas optionnels, ils sont OBLIGATOIRE. Si vous ne les mettez pas ou vous en mettez seulement 1, le 2068 va "planter".

-INPUT USR 62105,X,Y , remplis une région avec les attributs qu'on lui dit jusqu'au moment où elle rencontre un obstacle. X=(8*PAPER)+INK et Y=FLASH/BRIGHT comme vue ci-dessus. Tout le monde a déjà essayé de faire un rond plein en BASIC, c'est très long. Avec cette routine, c'est l'affaire de quelques secondes.
Faites CIRCLE 125,87,80 puis PLOT INVERSE 1,125,87 pour déterminer où le remplissage doit débuter. En assumant que le PAPER de l'écran est noir (sinon PAPER 0:RANDOMIZE USR 62048), on fait INPUT USR 62105,4,0 pour remplir notre cercle avec du vert (INK), car X=(8*0)+4=4.
Si on examine la liste ASSEMBLEUR de FILL, les 2 CALLs du début servent à évaluer les 2 paramètres suivant la fonction USR. A la sortie du deuxième CALL, le registre C contient X (PAPER&INK) et le registre B,Y (FLASH/BRGT)
Le CALL #2603 de la ligne 1010, appelle une routine de la ROM qui calcule l'adresse correspondante dans D_FILE1 à un PLOT x,y. Premièrement nous chargeons DE avec les coordonnées XY, nous appelons SCRML (#2603), au retour HL contient l'adresse dans D_FILE1 et l'accumulateur A contient la position du pixel dans l'octet. Ainsi si nous chargeons DE avec 0 et 175, i.e PLOT 0,175 qui se trouve en haut, à l'extrême gauche de l'écran, SCRML nous retourne HL=16384 et A=7.

-PRINT AT X,Y;INK/PAPER/BRGT/FLASH;CHR$ USR 62220 , affiche à l'écran la variable s$ avec des attributs différents de ceux en vigueur à l'écran.

```
10 LET s$="Ceci est un test..."
20 PAPER 1:INK 0:RANDOMIZE USR 62048:REM CLS
30 PRINT AT 10,5;FLASH 1;PAPER 0;INK 6;CHR$
   USR 62220
```

Pour accélèrer les choses, il serait bon de toujours initialiser la variable s$ au tout début d'un programme

```
1 LET s$="":LET c=3
...
100 LET s$=STR$ c:PRINT AT 20,0;BRIGHT 1;FLASH
   1;CHR$ USR 62220;" "
110 LET s$="TEST":PRINT PAPER 0;INK 4;CHR$ USR
   62220
```

Si par malheur, la variable s$ n'est definie lors de l'appel de la routine, alors le message d'erreur "Variable not found" sera affiché.

- RANDOMIZE USR 62292 , scroll D_FILE1 et D_FILE2 d'un caractère vers le haut. RANDOMIZE USR 62295 scroll seulement D_FILE2.

- RANDOMIZE USR 62346 , scroll D_FILE1 et D_FILE2 d'un caractère vers le bas. RANDOMIZE USR 62346 scroll seulement D_FILE2.

- RANDOMIZE USR 62446 , scroll D_FILE1 et D_FILE d'un caractère vers la gauche. RANDOMIZE USR 62461 pour D_FILE2 seulement.

- RANDOMIZE USR 62500 , scroll D_FILE1 et D_FILE2 d'un caractère vers la droite. RANDOMIZE USR 62515 pour D_FILE2 seulement.

NOTES SUPPLEMENTAIRES

Lorsque que D_FILE2 est en fonction et que le TOOLKIT est en place (n'oubliez pas le CLEAR 61999), il vous reste 28268 octets libres. Les UDGs débute maintenant à l'adresse 63256. Les programmes BASICs débutent eux à 31510. Donc pour faire une ligne 0, POKE 31511,0. Pour placer des codes machines dans des REMs, vous devez "pokez" à partir de 31514 en montant.

Pour sauver vos écrans sur cassette faites:

```
SAVE "D_FILE1"CODE 16384,6143
SAVE "D_FILE2"CODE 24576,6143
```

En tout temps, on peut retourner en mode normal avec OUT 255,0. On revient en EXTENDED COLOR MODE avec OUT 255,2.

Les routines de scrolls verticaux et horizontaux utilisent 32 octets de la mémoire-tampon de l'imprimante (BUFFER), i.e. entre l'adresse 23296 et 23552.

Lors de l'utilisation de FILL (INPUT USR 62105,X,Y), si on atteint le haut ou le bas de l'écran, le message d'erreur "Integer out of range" apparait. Dans un programme BASIC, l'utilisation de ON ERR GOTO et ON ERR RESET est le bienvenue pour remèdier à cette situation.

La commande DRAW fonctionne toujours mais nous avons pas de contrôle sur ses attributs, même chose pour CIRCLE.

En EXTENDED COLOR MODE, la zone ATTRIBUTE FILE2 est ouverte mais non utilisée, on peut l'utiliser pour emmagasiner de l'information ou des routines machines. Elle se situe entre 30720 et 31487, soit 767 octets.

PROGRAMMES DE DEMONSTRATION

```
 5 REM _____________DEMO1  GAG-o85
10 LET r=35: LET s=35: LET t=87
20 FOR n=0 TO 2* PI  STEP  PI /55
30 LET x=r* COS n: LET y=r* SIN n
35 LET papink=(8*0)+(( RND *6)+1): REM  PAPER =0 &  8< INK >0
40 PLOT x+s,y+t
45 INPUT  USR 62069,papink,0: REM couleur sur le pixel...
50 NEXT n
55 BEEP .1,-1: LET s$="APPUYEZ SUR UNE TOUCHE........."
60 PRINT  AT 20,0; BRIGHT 1; INK 4; CHR$  USR 62220: REM  PRINT s$...
70 PAUSE 0: FOR i=0 TO 31
80 RANDOMIZE  USR 62500: REM scroll -> droite
90 NEXT i
```

(annotation manuscrite : 6 PAPER 0: INK 0: RAND USR 62048)

```
  1 REM _________________________________DEMO2    GAG-o85
  5 BORDER 0: PAPER 0: INK 1: RANDOMIZE  USR 62048: REM   CLS
  9 REM   Dessine la maison
 10 PLOT 8,10
 20 DRAW 48,0: DRAW 0,48: DRAW -48,0: DRAW 0,-48
 30 PLOT 56,10: DRAW 70,20: DRAW 0,48: DRAW -70,-20
 40 PLOT 8,58: DRAW 24,30: DRAW 24,-30
 50 DRAW 70,20: DRAW -24,30: DRAW -70,-20
 60 PLOT 0,130: DRAW 175,-10: DRAW 80,10: DRAW 0,45: DRAW -255,0: DRAW 0,-45: REM   Le ciel
 70 PLOT 0,80: DRAW 30,30: DRAW 135,-5: DRAW 90,-25: REM  L'herbe
300 PLOT  INVERSE 1,15,15: INPUT  USR 62105,5,0: REM   FILL avec  PAPER 0 &  INK 5 (maison)
310 PLOT  INVERSE 1,57,20: INPUT  USR 62105,6,0: PLOT  INVERSE 1,122,59: INPUT  USR 62105,6,0
320 PLOT  INVERSE 1,32,71: INPUT  USR 62105,2,0
330 PLOT  INVERSE 1,175,150: INPUT  USR 62105,1,0: REM FILL (ciel)
340 PLOT  INVERSE 1,0,116: INPUT  USR 62105,4,0: REM FILL (herbe)
700 CIRCLE  INVERSE 1,180,150,10
```

```
  5 REM _________________________________DEMO3    GAG-o85
 10 BORDER 1: BRIGHT 1: PAPER 1: INK 7: RANDOMIZE  USR 62048
 20 CIRCLE 125,87,28
 30 PLOT  INVERSE 1,125,87: INPUT  USR 62105,14,1:
 40 CIRCLE  INVERSE 1,110,95,10
 50 CIRCLE  INVERSE 1,140,95,10
 60 PLOT 110,75: DRAW  INVERSE 1,30,0, PI
100 LET a$= INKEY$ : IF a$="" THEN  GO TO 100
110 IF a$="5" THEN  RANDOMIZE  USR 62446
120 IF a$="8" THEN  RANDOMIZE  USR 62500
130 IF a$="7" THEN  RANDOMIZE  USR 62292
140 IF a$="6" THEN  RANDOMIZE  USR 62346
150 GO TO 100
```

suite p. suivante

(suite de la page précédente)

L

```
00010 ;************************
00020 ;*TOOLKIT ULTRA RES 2068*
00030 ;* GAG-o       JUILLET 85*
00040 ;*     version  3.5     *
00050 ;************************
00060 ;
00070          ORG 62000
00080          DISP 53536
00090 ;________________________
00100 ;Switch D_FILE 2
00110 ;->62000d  point d'entree
00120 ;________________________
00130 SW     LD BC,65278
00140        CALL 25753 ;EXROM
00150        LD A,2     ;MODE 2
00160        CALL 3726  ;CH_VID
00170        LD BC,255
00180        CALL 64601 ;HOME
00190        CALL CLS
00200 MOT    LD HL,TEXT ;message
00210        LD B,64    ;du
00220 LOOP1  LD A,(HL)  ;debut
00230        RST #10
00240        INC HL
00250        DJNZ LOOP1
00260        LD HL,26400
00270        CALL SOU
00280        LD HL,24576
00290 SOU    LD B,32    ;sou-
00300        LD A,128+7 ;ligne
00310 LOOP2  LD (HL),A  ;avec
00320        INC HL     ;FLASH
00330        DJNZ LOOP2
00340        RET
00350 ;________________________
00360 ;Effacement d'ecran
00370 ;->62048d  point d'entree
00380 ;________________________
00390 CLS    CALL #8E4  ;CLSROM
00400 ;________________________
00410 ;->62051d  point d'entree
00420 ;________________________
00430 CLS2   LD A,(23693)
00440        LD (24576),A
00450        LD HL,24576;D_F2
00460        LD DE,24577
00470        LD BC,6144
00480        LDIR
00490        RET
00500 ;________________________
00510 ;Attribut sur PLOT
00520 ;->62069d  point d'entree
00530 ;________________________
00540 ATTR   CALL 7132  ;evalue
00550        CALL 9824  ;param.
00560         ;reg.C=ink+8*paper
00570         ;reg.B=flash+brght
00580        LD A,B
00590        CP 0
00600        JR Z,ATTRP
00610         ;B=000000XX C
00620        RR B
00630        RR B
00640        RR B
00650        AND #C0
00660         ;B=XX000000 0
00670 ATTRP  PUSH BC
00680        LD BC,(23677)
00690        CALL #2603 ;SCRML
00700             ;HL->D_FILE1
00710        LD BC,8192
00720        ADD HL,BC
00730             ;HL->D_FILE2
00740        POP BC
00750        LD A,B
00760        ADD A,C
00770        LD (HL),A  ;couleur
00780        RET
00790 ;________________________
00800 ;           FILL
00810 ;->62105d  point d'entree
00820 ;________________________
00830 FILL   CALL 7132  ;evalue
00840        CALL 9824  ;param.
00850        CALL ARGU
00860        LD DE,(23677);coord
00870 HBYTE  CALL BYTE
00880        JR NZ,RPIX
00890        CALL DBYTE  ;x=x+1
00900        INC D       ;y=y+1
00910        JR HBYTE
00920 RPIX   LD DE,(23677)
00930        DEC D
00940 BBYTE  CALL BYTE
00950        RET NZ
00960        CALL DBYTE
00970        DEC D       ;y=y-1
00980        JR BBYTE
00990 BYTE   LD B,D
01000        LD C,E
01010        CALL #2603  ;SCRML
01020        LD B,A
01030        INC B
01040        LD C,B
01050        LD A,(HL)
01060 BYTE1  RLCA
01070        DJNZ BYTE1
01080        BIT 0,A     ;pix.1?
01090        RET NZ      ;1 RET
01100        SET 0,A     ;0->1
01110        LD B,C
01120 BYTE2  RRCA
01130        DJNZ BYTE2
01140        LD (HL),A   ;PLOT
01150        LD BC,8192
01160        ADD HL,BC
01170        LD A,(ATTPL)
01180        LD (HL),A   ;attri.
01190        RET
01200 DBYTE  INC E       ;x=x+1
01210        CALL BYTE   ;verif.
01220        JR NZ,RPIX2
01230        JR DBYTE
01240 RPIX2  LD A,(23677);restor
01250        LD E,A      ;e x&y
01260 GBYTE  DEC E       ;x=x-1
01270        CALL BYTE   ;verif.
01280        JR NZ,RPIX3
01290        JR GBYTE
01300 RPIX3  LD A,(23677)
01310        LD E,A
01320        RET
01330 ATTPL  DEFB 0      ;attri.
01340 ARGU   LD A,B      ;FLASH-
01350        CP 0        ;BRGHT?
01360        JR Z,ARG1
01370        RR B
01380        RR B
01390        RR B
01400        AND #C0
01410 ARG1   LD A,B
01420        ADD A,C
01430        LD (ATTPL),A
01440        RET
01450 ;________________________
01460 ;Attribut avec PRINT
01470 ;->62220d  point d'entree
01480 ;________________________
01490 PR     LD HL,(#534B);VARS
01500        LD BC,(#5C59);ELINE
01510 U1     PUSH HL
01520        SBC HL,BC
01530        JR Z,MEM
01540        POP HL
01550        LD A,(HL)
01560        CP 83       ;CODE"S"
01570        JR Z,VERI
01580        INC HL
01590        JR U1
01600 MEM    RST 8       ;S$ EST
01610        DEFB 1      ;PAS LA
01620 VERI   INC HL
```

suite ...

(suite de la page precedente)

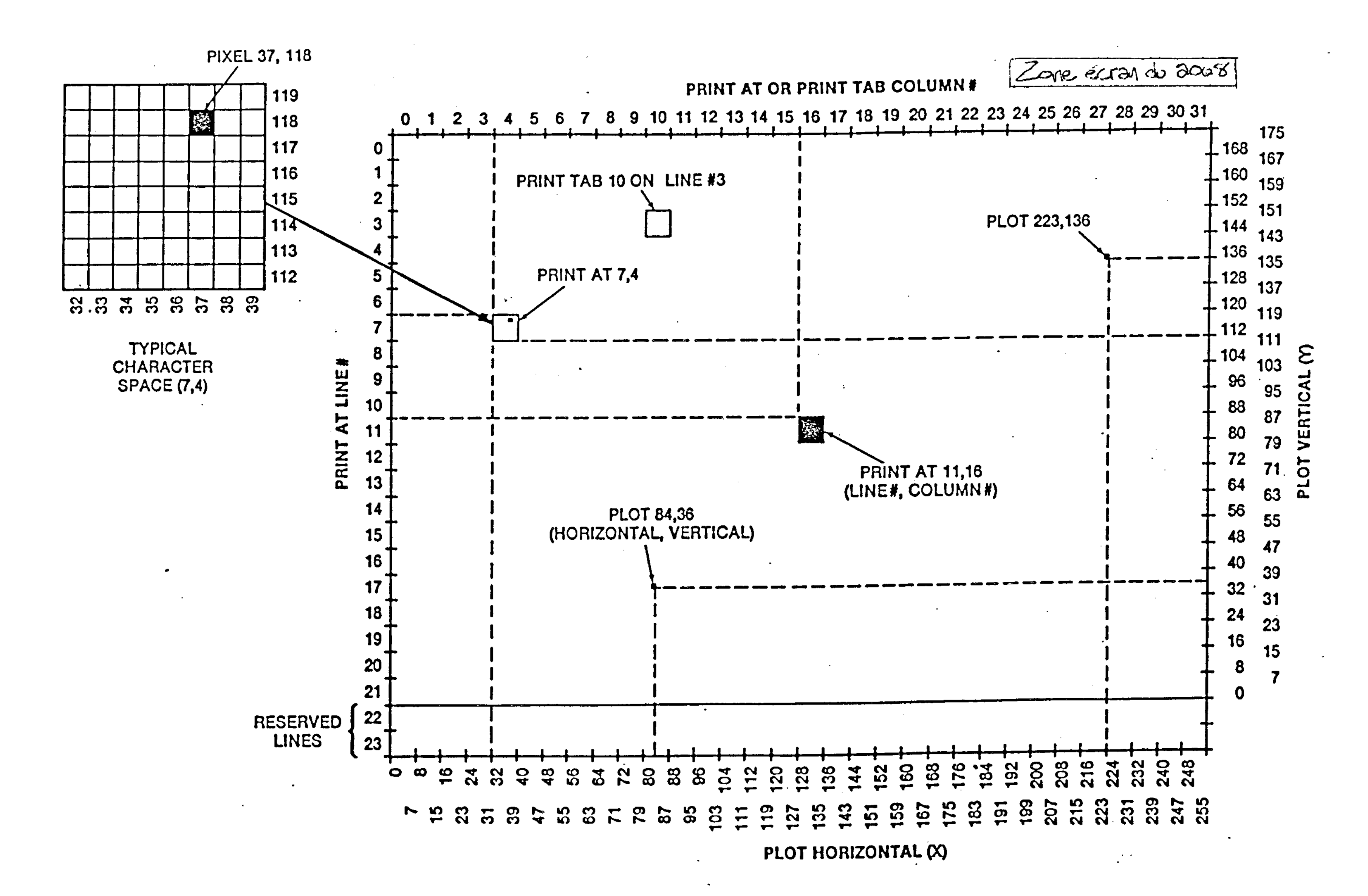
```
01630        INC HL
01640        LD A,(HL)
01650        CP 0
01660        JR Z,PRS
01670        INC HL
01680        JR U1
01690 ERR    RST 8        ;retour
01700        DEFB 1       ;BASIC
01710 PRS    DEC HL
01720        LD B,(HL)     ;LENS$
01730        INC HL
01740 U2     INC HL
01750        PUSH BC
01760        CALL PRAT
01770        POP BC
01780        LD A,(HL)
01790        RST #10
01800        DJNZ U2
01810        LD A,8        ;Cur.
01820        RST #10       ;LEFT
01830        LD C,(HL)
01840        RET
01850 PRAT   PUSH HL
01860        LD HL,(23684);DFCC
01870        LD A,(23695) ;ATTT
01880        LD DE,8192
01890        ADD HL,DE     ;D_F2
01900        LD B,8
01910 PRAT1  LD (HL),A     ;attr.
01920        INC H         ;+256
01930        DJNZ PRAT1
01940        POP HL
01950        RET
01960 ;________________________
01970 ;SCROLL vers le HAUT
01980 ;->62292d  point d'entree
01990 ;________________________
02000 SCRH1 CALL 2361     ;ROMSCR
02010 ;________________________
02020 ;->62295d  point d'entree
02030 ;________________________
02040 SCRH2 LD HL,#6000 ;D_F2
02050       LD DE,#5BE0 ;BUFFER
02060 H1    PUSH DE
02070       PUSH HL
02080       LD A,3
02090       LD BC,#20     ;32d
02100 H2    PUSH BC
02110       PUSH HL       ;D_F2->
02120       LDIR          ;BUFFER
02130       POP DE
02140       LD C,#E0      ;scroll
02150       LDIR          ;D_F2
02160       LD B,7
02170       ADD HL,BC
02180       DEC A
02190       POP BC
02200       JR NZ,H2
02210 H3    LD B,C
02220       LD A,(23693);ATT_T
02230 H4    LD (DE),A
02240       INC DE
02250       DJNZ H4
02260 H5    POP HL
02270       POP DE
02280       INC H
02290       LD A,H
02300       CP #68
02310       JR C,H1
02320 H6    EX DE,HL      ;met le
02330 H7    LD (HL),E     ;BUFFER
02340       INC L         ;a 0
02350       JR NZ,H7
02360       RET
02370 ;________________________
02380 ;SCROLL  vers le BAS
02390 ;->62346d  point d'entree
02400 ;________________________
02410 SCRB1 LD HL,#57FF ;D_F1
02420       LD DE,#5BFF ;BUFFER
02430       LD A,0        ;mettre
02440       LD BC,VAR2    ;a 0 la
02450       LD (BC),A     ;ligne1
02460       JR SCRB
02470 ;________________________
02480 ;->62360d  point d'entree
02490 ;________________________
02500 SCRB2 LD HL,#77FF ;D_F2
02510       LD DE,#5BFF ;BUFFER
02520       LD A,(23693);mettre
02530       LD BC,VAR2    ;ATT_P
02540       LD (BC),A     ;ligne1
02550 SCRB  PUSH DE       ;meme
02560       PUSH HL       ;princi
02570       LD A,3        ;que
02580       LD BC,#20     ;scroll
02590 B1    PUSH BC       ;vers
02600       PUSH HL       ;le
02610       LDDR          ;haut
02620       POP DE
02630       LD C,#E0
02640       LDDR
02650       LD B,#F9
02660       ADD HL,BC
02670       POP BC
02680       DEC A
02690       JR NZ,B1
02700 B2    LD B,C
02710       LD A,(VAR2)
02720 B3    LD (DE),A
02730       DEC DE
02740       DJNZ B3
02750       POP HL
02760       POP DE
02770       DEC H
02780       LD A,H
02790       PUSH AF      ;est-ce
02800       LD A,(VAR2);DF1 ou
02810       CP 0         ;DF2
02820       JR NZ,BB4
02830       POP AF       ;D_F1
02840       CP #50       ;limite
02850       JR NC,SCRB
02860       XOR A
02870       LD B,#20
02880 B4    LD (DE),A
02890       DEC DE
02900       DJNZ B4
02910       JR SCRB2
02920 BB4   POP AF       ;D_F2
02930       CP #70       ;limite
02940       JR NC,SCRB
02950       XOR A
02960       LD B,#20
02970 BBB4  LD (DE),A
02980       DEC DE
02990       DJNZ BBB4
03000       RET
03010 ;________________________
03020 ;SCROLL   vers la GAUCHE
03030 ;->62446d  point d'entree
03040 ;________________________
03050 VAR1  DEFW 0
03060 VAR2  DEFB 0
03070 ;
03080 SCRG1 LD HL,16385 ;D_F1
03090       LD DE,16384
03100       LD A,0
03110       LD BC,VAR2
03120       LD (BC),A
03130       CALL SCRG
03140 ;________________________
03150 ;->62461d point d'entree
03160 ;________________________
03170 SCRG2 LD HL,24577 ;D_F2
03180       LD DE,24576
03190       LD A,(23693)
03200       LD BC,VAR2
03210       LD (BC),A
03220 SCRG  LD BC,192
03230 G1    LD (VAR1),BC
03240       LD BC,31
03250       LDIR
03260       LD A,(VAR2)
03270       LD (DE),A
```

(suite de la page précédente)

```
03280        INC DE
03290        INC HL
03300        LD BC,(VAR1)
03310        DEC C
03320        JR NZ,G1
03330        RET
03340 ;_______________________
03350 ;SCROLL  vers la DROITE
03360 ;->62500d  point d'entree
03370 ;_______________________
03380 SCRD1 LD HL,22526 ;D_F1
03390        LD DE,22527
03400        LD A,0
03410        LD BC,VAR2
03420        LD (BC),A
03430        CALL SCRD
03440 ;_______________________
03450 ;->62515d  point d'entree
03460 ;_______________________
03470 SCRD2 LD HL,30718 ;D_F2
03480        LD DE,30719
03490        LD A,(23693)
03500        LD BC,VAR2
03510        LD (BC),A
03520 SCRD  LD BC,192
03530 D1    LD (VAR1),BC
03540        LD BC,31
03550        LDDR
03560        LD A,(VAR2)
03570        LD (DE),A
03580        DEC DE
03590        DEC HL
03600        LD BC,(VAR1)
03610        DEC C
03620        JR NZ,D1
03630        RET
03640 ;_______________________
03650 ;TEXTE de BIENVENUE
03660 ;_______________________
03670 TEXT   DEFM /ULTRA HIGH /
03680        DEFM /COLOR RESOL/
03690        DEFM /UTION 2068/
03700        DEFM /TOOLKIT/
03710        DEFM /     /
03720        DEFM /GAG-o   /
03730        DEFM / version 3.5/
03740 ;_______________________
03750 FIN DEFB 0
```

Liste: HEX-LOADER   TOOLKIT ULTRA-RES 2068

```
  5 CLEAR 61999: LET a=10: LET b=11: LET c=12: LET d=13: LET e=14: LET f=15
 10 LET ligne=100: LET adresse=62000
 15 READ s$,somme: LET tot=0
 20 LET byte=16* VAL s$(1)+ VAL s$(2): LET tot=tot+byte: POKE adresse,byte
 25 LET s$=s$(3 TO ): LET adresse=adresse+1: IF s$ <> "" THEN GO TO 20
 30 POKE 23692,255: IF somme=tot THEN  PRINT "Ligne ";ligne;" O.K.": LET ligne=ligne+1: GO TO 15
 40 BEEP .1,1: PRINT "Erreur a la ligne ";ligne: STOP
100 DATA "01FEFECD99643E02CD8E0E01FF00CD59FCCD60F2215AF406407ED72310FB212067CD57F22100600620 3E87772310FCC9CDE4083A8D5C32006021006D11016001",6628
101 DATA "0018EDB0C9CDDC1BCD602678FE002808CB18CB18CB18E60C0C5ED4B7D5CCD032601002009C1788177C9CDDC1BCD6026CDF9F2ED5B7D5CCDC0F22006CDDFF21418",8153
102 DATA "F5ED5B7D5C15CDC0F2C0CDDFF21518F6424BCD03264704487E0710FDCB47C0CBC7410F10FD7701002009 3AF8F277C91CCDC0F2200218F83A7D5C5F1DCDC0F220",7828
103 DATA "0218F83A7D5C5FC90078FE002808CB18CB18CB18E6C078913 2F8F2C92A4B53ED4B595CE5ED422809E17EFE5328052318F2CF0123237EFE002805231 8E6CF012B",6911
104 DATA "462323C5CD41F3C17ED710F63E08D74EC9E52A845C3A8F5C110020190608772410FCE1C9CD3909210060 11E05BD5E53E03012000C5E5EDB0D10EE0EDB0060709",6928
105 DATA "3DC120F0413A8D5C121310FCE1D1247CFE6838D9EB7320 20FCC921FF5711FF5B3E0001EDF302180D21FF7711FF5B3A8D5C01EDF302D5E53E03012000C5E5EDB8",7542
106 DATA "D10EE0EDB806F909C13D20F0413AEDF3121B10FCE1D1257CF53AEDF3FE0D200EF1FE5030D0AF06201 21B10FC18BAF1FE7030C2AF0620121B10FCC90000002101",7429
107 DATA "401100403E0001EDF302CD0AF42101601100603A8D5C01EDF30201C000ED43EBF3011F00EDB03AEDF3121323ED4BEBF30D20EAC921FE5711FF573E0001EDF302",6744
108 DATA "CD40F421FE7711FF773A8D5C01EDF30201C000ED43EBF3011F00EDB83AEDF3121B2BED4BEBF30D20EAC9554C54524120484947482043 4F4C4F52205245534F4C",6956
109 DATA "555449 4F4E2032303638544F4F4C4B4954202020202047414 72D6F2020202076657273696F6E20332E350000000000000000000000000000000000000000000",2796
500 REM _______________________________Sauve l'HEXLOADER
510 SAVE "ultra_1": PRINT "Je verifie ": VERIFY "": PRINT "OK.": STOP
550 REM _______________________________Sauve les CODES
560 SAVE "ultra_c" CODE 62000,620: PRINT "Je verifie ";: VERIFY "" CODE : PRINT "OK.": STOP
```



PIXEL 37, 118
TYPICAL CHARACTER SPACE (7,4)
PRINT AT OR PRINT TAB COLUMN #
PRINT AT LINE #
PRINT TAB 10 ON LINE #3
PRINT AT 7,4
PRINT AT 11,16 (LINE#, COLUMN#)
PLOT 84,36 (HORIZONTAL, VERTICAL)
PLOT 223,136
RESERVED LINES
PLOT HORIZONTAL (X)
PLOT VERTICAL (Y)
Zone écran du 2068

SINCLAIR QL



S I N C L A I R   Q L

* 32 bits
* Memoire de 128 K
* 2 Microdrives
* Haute resolution couleur
* 4 Logiciels inclus

- CONFIGURATION DU SYSTEME

Le systeme de base ($ 799.95) inclus deux microprocesseurs, 128 K de memoire , un systeme d'exploitation, un langage de programmation, un clavier QWERTY de 65 touches, 2 microdrives d'une capacite de 100 K chaque, 4 logiciels sur microdrive, 4 microcassettes vierges, source d'alimentation , les cables et les manuels.

- HARDWARE

RAM: 128 K, peut etre augmente a 640 K. 32 K pour memoire-ecran.

ROM: 32 K, contient Sinclair SuperBASIC et Sinclair QDOS, le systeme d'exploitation.

CPU: MOTOROLA 68008, 7.5 Mhz + INTEL 8049

VIDEO: Graphique haute resolution en 2 modes:
Mode 1- 512x256 pixels (4 couleurs)
Mode 2- 256x256 pixels (8 couleurs)
De 40 a 85 caracteres par ligne.

ALIMENTATION: 120 V, 60 Hz, 9 Vcc a 1.8 A, 15.6 Vac a .2 A du transformateur. (19 watts a l'ordinateur)

DIMENSION: 138 mm x 46 mm x 472 mm

POIDS: 1388 g (3.055 lbs.)

LOGICIEL

-SYSTEME D'EXPLOITATION

QDOS: Possibilites de "MULTI-TASKING"
Schedule de priorite d'execution
Fenetres (Windows)

QL QUILL: Traitement de texte

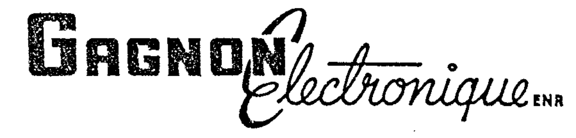
QL ABACUS: Tableur (Spreadsheet)

QL ARCHIVE: Base de donnees.

QL EASEL: Graphique

INTERFACES

-Trappe pouvant recevoir jusqu'a 32 K de ROM supplementaire.
-2 prises JOYSTICKS.
-Trappe pouvant recevoir jusqu'a .5 Mb de RAM supplementaire.
-2 RS-232-C ,de 75 a 19200 bauds a la transmission et recoit jusqu'a 9600 bauds.
-Prise TV
-Prise moniteur, RGB et monochrome.
-QLAN permet de relier 64 QL ensemble.



tél.: 527·6103

390, BOUL. CENTRAL NORD DUBERGER   QUÉBEC   G1P 3P9